DOCUMENT RESUME

ED 061 976 52 LI 003 646

AUTHOR	Silver, Steven S.
TITLE	FMS: A Format Manipulation System for Automatic Production of Natural Language Documents, Second Edition. Final Report.
INSTITUTION	California Univ., Berkeley. Inst. of Library Research.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-7-1083
PUB DATE	Sep 71
GRANT	OEG-1-7-071083-5068
NOTE	46p.; (7 References)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Character Recognition; *Electronic Data Processing; *Information Processing; *Machine Translation; Manuals; *Programing Languages
IDENTIFIERS	Berkeley; FMS; Format Manipulation System; *University of California

ABSTRACT

FMS/3 is a syscem for producing hard copy documentation at high speed from free format text and command input. The system was originally written in assembler language for a 12K IBM 360 model 20 using a high speed 1403 printer with the UCS-TN chain option (upper and lower case). Input was from an IBM 2560 Multi-function Card Machine. The model 20 version has been superceded by a revised implementation operational on all other models of the IBM System/360. Increased speed is gained by restricting performance to text formatting only, eliminating editing and text scanning functions, such as spelling checking, index generation, etc. This document serves as language specifications and users manual and was completely generated by FMS. It contains an introduction to the system, including the implementation and project history; a chapter on using FMS; an FMS beginners cookbook, and an explanation of Model 20 tape file management. [Related documents are LI 003610, LI 003611, LI 003645, LI 003647 and LI 003648.] (Author/SJ)

BR-7-1083
PA-52

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

FINAL REPORT
Project No. 7-1083
Grant No. OEG-1-7-071083-5068

ED 061976

FMS: A FORMAT MANIPULATION SYSTEM FOR AUTOMATIC PRODUCTION OF NATURAL LANGUAGE DOCUMENTS*

Second Edition

By

Steven S. Silver

Institute of Library Research
University of California
Berkeley, California 94720

September 1971

The research reported herein was performed pursuant to a grant with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

LI 003 646

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

*Second edition, date produced: December, 1969.

TABLE OF CONTENTS

ABSTRACT..... 2
I INTRODUCTION..... 3
IMPLEMENTATION..... 3
PROJECT HISTORY..... 3
II USING FMS..... 5
BASIC CONCEPTS..... 5
TERMINOLOGY..... 6
CONTROL COMMANDS..... 7
THE LINE GROUP..... 7
THE PROGRAM STATUS GROUP..... 8
CASE CONTROL..... 9
FORCING NEW LINES..... 9
Paragraphing..... 10
Line Termination..... 10
OVERALL PAGE CONTROL..... 11
Line Spacing..... 11
Page Numbering..... 12
New Pages..... 12
Page Extent..... 13
LINE CONTROL..... 13
Line Justification..... 14
Underlining..... 15
Boldfacing..... 15
Paragraph Indentation..... 15
Tabbing..... 16
Column Skipping..... 16
LINE SELECTION..... 17
PROGRAM STATUS GROUP CONTROL..... 18
III FMS BEGINNERS COOK BOOK..... 19
IV MACROS..... 21
V MODEL 20 TAPE FILE MANAGEMENT..... 23
APPENDIX I: COMMAND LIST..... 24
Key to Syntax..... 24
Commands..... 25
APPENDIX II: CHARACTER SET..... 33
APPENDIX III: DEFAULT FORMAT CONDITIONS..... 34
APPENDIX IV: ERROR MESSAGES..... 35
APPENDIX V: MODEL 20 OPERATION..... 37
Minimum System Requirement..... 37
Loading..... 37
Halt Codes..... 37
APPENDIX VI: EXAMPLES..... 38
APPENDIX VII: FMS JOB CONTROL LANGUAGE AT UCLA.. 42
APPENDIX VIII: SPEED AND COST..... 43

FOREWORD

This report contains the results of the second phase (July, 1968 - June, 1970) of the File Organization Project, directed toward the development of a facility in which the many issues relating to the organization and search of bibliographic records in on-line computer environments could be studied. This work was supported by a grant (OEG-1-7-071083-5068) from the Bureau of Research of the Office of Education, U.S. Department of Health, Education, and Welfare and also by the University of California. The principal investigator was M.E. Maron, Professor of Librarianship and Associate Director, Institute of Library Research; the project director and project manager were, respectively, Ralph M. Shoffner and Allan J. Humphrey, Institute of Library Research.

This report is being issued as seven separate volumes:

- Shoffner, Ralph M., Jay L. Cunningham, and Allan J. Humphrey. The Organization and Search of Bibliographic Records in On-line Computer Systems: Project Summary.

- Shoffner, Ralph M. and Jay L. Cunningham, eds. The Organization and Search of Bibliographic Records: Component Studies.

- Aiyer, Arjun K. The CIMARON SYSTEM: Modular Programs for the Organization and Search of Large Files.

- Silver, Steven S. INTX: Interactive Assembler Language Interpreter Users' Manual.

- Silver, Steven S. FMS: Users' Guide to the Format Manipulation System for Natural Language Documents.

- Silver, Steven S. and Joseph C. Meredith. DISCUS Interactive System Users' Manual.

- Smith, Steven F. and William Harrelson. TMS: A Terminal Monitor System for Information Processing.

Because of the joint support provided by the Information Processing Laboratory Project (OEG-1-7-071085-4286) for the development of DISCUS and of TMS, the volumes concerned with these programs are included as part of the final report for both projects. Also, the CIMARON system (which was fully supported by the File Organization Project) has been incorporated into the Laboratory operation and therefore, in order to provide a balanced view of the total facility obtained, the volume is included as part of the Laboratory project report. (See Maron, M.E. and Don Sherman, et al. An Information Processing Laboratory for Education and Research in Library Science: Phase 2. ILR 1971.)

ABSTRACT

FMS/3 is a system for producing hard copy documentation at high speed from free format text and command input.

The system was orginally written in assembler language for a 12K IBM 360 model 20 using a high speed 1403 printer with the UCS-TN chain option (upper and lower case). Input was from an IBM 2560 Multi-function Card Machine. The model 20 version has been superceded by a revised implementation operational on all other models of the IBM System/360.

Increased speed is gained by restricting performance to text formatting only, eliminating editing and text scanning functions. (i.e. spelling checking, index generation, etc.).

This document serves as language specifications and users manual and was completely generated by FMS.

I. INTRODUCTION

FMS/3 is a system for producing hard copy documentation at high speed from free form text and command input. It operates under OS 360 on an IBM 360 model 91 and has a subset usable on an IBM 360 model 20.

This document serves as language specifications and users manual and was completely generated by FMS. The text was reviewed and criticized by members of the Institute of Library Research, The Campus Computing Network and selected individuals. Special thanks is extended to Steve Barryte, Vernon Coley III, Bruce Dane, and Peter Watson who edited and modifed the machine readable source text.

IMPLEMENTATION

Version 3 of FMS was originally written in IBM 360 model 20 assembler language, with an assembler developed by the University of Waterloo, Ontario, Canada, used on an IBM 360 model 75. The object deck is 16K in length. Model 20 FMS uses a queued I/O method developed by Howard B. Golden of the UCLA Computer Club resulting in a 40% increase in model 20 processing speed. The full 12K model 20 system uses a tape access method developed by Van Martin of the Campus Computing Network, modified by Tom Springer, also of the Network. The model 20 version has been rendered obsolete by the full FMS described in this document.

PROJECT HISTORY

The FMS project started in late 1965, based on discussions held by members of the UCLA Computer Club. The University had installed an IBM 1401 computing system, modified to handle a 120 character print chain, and some initial programming established the feasibility of a text processing system for the machine. One system (developed by James Lewis) had extensive format capability but was severely limited by computational overhead. The other system, FMS/1, was considerably faster but not nearly as powerful. In both cases the limitation was found to be the

computational speed and the 4K memory capacity of the 1401.

Interest in text processing was again revived when a special project on text manipulation (Mathematics 109 under Professor David Pope at UCLA) was completed in 1966, and by the re-implementation of FMS by Stephen Wolfe for the University's 360 model 75 computer. The speed of execution of this implementation more than made up for limitations in the language as specified at that time. Also contributing to interest were two IBM distributed packages: TEXT/360 and FORMAT/44, both comprehensive in scope but still limited by execution speed and the large amount of storage required.

The revised version of FMS, FMS/3, was released for testing in summer 1968. It was designed to take advantage of the speed and availability of the 360 model 20 computing system, and the experience gained in using some of the other text processing systems. By fall 1969 the most readily available computing system was the 360/91 which encouraged the development of this latest version designed to run on any true 360 and use model 20 for off-line printing of the final results.

II. USING FMS

BASIC CONCEPTS

FMS is designed to deal with a continuous stream of text interspersed with control commands. Input is in card image form, all eighty columns of which may be used. Column 80 immediately precedes column 1 of the next card. If no control commands are issued, the input stream is treated as lower case text which will fall into a format defined by a Program Status Group. Each line begins at the left margin and ends at a word boundary (the start of the next word) at or before the right margin. When enough text is supplied to fill a page, a new page is automatically started.

A "/" before a letter changes the case of the letter, hence lower case letters are capitalized and upper case letters are made lower case. To print a "/" the string "//" is used.

Example:

```
/aa/bb/11/&&// /and /so /fort/h --->

AaBb!1+&/ And So FortH
```

Blanks are reduced to one blank whenever more than one follow in sequence.

Example:

```
this    is     shorter    than   it  seems
--->
this is shorter than it seems
```

Certain characters are treated in special ways to allow text format control. The characters ".", "?", and "!" are unique since the number of blanks following these characters will not exceed two, but can number one or zero.

Example:

```
end?/begin          ---> end?Begin
end? /begin         ---> end? Begin
end?  /begin        ---> end?  Begin
end?   /begin       ---> end?  Begin
```

An "@" indicates that the next two characters are format control commands as described under CONTROL COMMANDS. To print an "@" the command "@@@" must be issued.

Blanks between commands are not ignored. If you find a troublesome blank that refuses to disappear, examine residual blanks between commands or between commands and text.

See the section "FMS Beginners Cook Book" for a simple use of the FMS system.

TERMINOLOGY

To make it easier to explain concepts in FMS, the following terms are useful:

line A single printable line in FMS. It may have text in it, it may be completely blank, or it may be declared non-existent (physically existing but not printable). The basic routines that make up FMS usually deal with each line on an individual basis. A line may have up to 132 usable print positions.

line group A set of lines consisting of a superscript, main, overprint, boldface, and subscript line with carriage control. When FMS finds that one of the lines has been filled it will try to print the whole line group. Line groups print only if there is something in the main line. See the example under The Line Group.

line in question The line that FMS is currently dealing with. The subscript, superscript, overprint, and main line may be a line in question.

line output pointer The pointer to a specific character or column in a line in question. Any position within a line group can be described by the combination of the line in question and the line output pointer.

macro A method by which commands and text may be

filed away in a manner that permits their reinsertion into the input stream on command. Frequently used text and format controls may be simplified by using this technique.

CONTROL COMMANDS

There will certainly be times when the standard flow of text must be disrupted for line or page format changes. Page skipping, paragraphing, column or line spacing, etc. all fall into this category.

To deviate from default format conditions, the user must supply overriding format controls of the form @nn, where nn is the particular control to be exercised. Control commands are available to modify the Program Status Group (PSG), change any member of the line group, and control spacing and local format. The rest of this section will discuss, in detail, FMS commands controlling these conditions. Appendix I is a complete list of all commands together with a synopsis of their functions. Macro commands will be treated in a special section.

Example:

```
@PG eject a page
@SK004 skip four lines
@MU4 use macro 4
```

THE LINE GROUP

The basic logical printing unit produced by FMS is called a line group. It is created by the text and command input supplied in the input stream. A line group consists of a superscript line, an overprint line, a boldface line, a main line, a subscript line and carriage spacing control.

Each line of the line group has associated with it a hidden existence flag indicating whether or not any information (including explicit blanks) has been placed in the line, and printer carriage control (e.g. print with space suppress, print with single space, etc.).

Also attached to each of these lines is an overflow sub-line, used to save information started but not completed on its associated print line. This overflow line holds data which has not printed because of the line termination

process. The overflow is inserted in the line group immediately following that which has just been printed.

When the command is given to dump a line, either explicitly by command or implicitly by line overflow, the dump routine attempts to print out all lines. If the main line does not exist, then nothing is printed. If no superscripting or subscripting has been done in the past, the sub and superscript lines are not printed. If subscript or superscript lines have been used before, but are currently unused, they will be represented by a blank line to satisfy format requirements (giving the general appearence of triple spacing). The overprint line is printed next (if it exists), without moving the carriage along with multiple strikes of the boldface line, followed by the main line.

EXAMPLE:

```
 2    2    2
X  = Y  + Z  equation 1   main, overprint, boldface
 i    i    i              subscript
```

When the whole group has been completely printed, normal carriage control is taken -- one less line is skipped than the number indicated in the last @CC command (i.e., @CC2 skips one line between line groups).

NOTE: In the 91 implementation all internal carriage control is 360 machine dependant and is optimized and converted to standard USASI codes before printing.

THE PROGRAM STATUS GROUP

Changing the overall format of FMS output is done by modifying the Program Status Group (PSG). This collection of counters and constants control the general look of an output page (i.e., margins, tabbing, paragraph indentation, etc.).

As each character is received in the input stream it is placed in an output line buffer. The boundaries for a character's placement are stored in and controlled by the PSG. Except in the case of changes to the right and left margins, which go into effect after the current line group is printed, changes to the PSG go into effect immediately.

Changes to the PSG are made by issuing special control

commands. A list of default values of the PSG can be found in Appendix III, Default Format Conditions.

The PSG contains Current values for the following:

Sub- Superscript switch
Upper and Lower Case switch
Text Body Length
Paragraph Indentation
Left Margin
Carriage Spacing
Underlining switch
Bold Face switch
Margin Justification switch
Tab Stops
Header Spacing
Right Margin
Underlining Character
Quote Mode switch
Line Justification switch

CASE CONTROL

A "/" before a character will capitalize that character if in lower case mode, or change that character to lower case if in upper case mode. In accordance with the table in Appendix II, non-alphabetic characters become special symbols. Long character strings may be capitalized by using the command @UC before and @LC after the string to restore normal lower case.

Example:

```
/a/b/c/de ---> ABCDe
@UCabcd@LCe ---> ABCDe
@UCsweat /a/n/d tears@LC ---> SWEAT and TEARS
```

FORCING NEW LINES

Certain conditions will arise when termination of a line other than by normal line overflow is desired. This may be done by using the following commands:

Paragraphing

The "new paragraph" command (@PP) will force the printing of the current line, even though not extended to the right margin. The text following the command will start at the current value of the paragraph indentation location, as set by the @IS command. If the value of the indentation is the same as the value of the left margin, (set by an @LM) an extra blank line is placed between paragraphs in addition to normal carriage control. The @CC carriage control command has a special setting (0) which gives normal single spacing but will leave an extra blank line between paragraphs regardless of indentation location. See also paragraph indentation in this section.

Example:

```
@IS035@LM033@RM055
                    This is an example
                  of margin control.
                    The use of the @PP
                  makes     paragraphing
                  easy.
                    @LM035

                    Block paragraphs are
                    possible by setting
                    the left margin
                    equal to the
                    indentation.

                    This is the way it
                    looks.
```

Line termination (dump buffer)

The @DB and @DE commands force the termination of any line group in progress. It is similar to the new paragraph command, @PP, except that the line position pointer is set equal to the left margin, and no extra lines, other than normal carriage control, are inserted. If the line group in progress had no text in it, nothing will be printed; thus multiple @DB's will, at most, force the termination of only one line.

The line dump commands are used when only a limited amount of text is required on a printed line. If even margins are in effect, the use of the @DB command, or any

command that forces a new line, will force the current line group being printed to be left justified, producing better-looking results. If even margins are required the @DE command can be used, dumping the line group just as if the right margin had been passed.

Example:

```
@EM
 @DBdump@DBbuffer@DBworks like@DBthis --->

dump
buffer
works like
this
  --------------------
 @DEdump@DEbuffer@DEeven@DEworks like@DEthis ---

dump
                                                    buffer

even
works                                                 like
this
```

OVERALL PAGE CONTROL

Three commands are available to control page skips; the unconditional page skip command (@PG), the conditional page skip command (@PS), and the carriage control command (@CC). Normal text flow will cause pages to be skipped automatically at the correct time.

Line spacing

Normal carriage spacing may be varied through the use of the @CC command, which controls spacing between lines. Lines may be skipped by issuing an @SP command, which either skips a certain number of lines, or goes to the top of the next page, whichever is less or by issuing an @SK, which spaces the number of lines specified, ignoring page boundaries.

Example:

```
 @DB1@SK0022@SP0023@DB --->
1

2

3
```

Page Numbering

Pages are normally sequentially numbered starting at page 0 allowing the user to request a new page (page 1) to properly start his report. There is no certainty that PMS will initially start printing at the top of the first page unless a specific command request is made. The @PB### and @PN### commands allows you to change the logical number of the current page causing subsequent pages to be appropriately numbered. If "###" is entered as "***" then page numbering is suppressed.

Example:

@PN*** ---> page numbering is suppressed.
@PN012 ---> the current page is logically page 12. This will be reflected starting on the top right of the next page (13).
@PB012 ---> the current page is logically page 12 reflected by a page number centered at the bottom of the page.

Horizontal page number positioning is controlled by the settings of the absolute right and left margins.

New pages

If a new page is to be started immediately, an @PG may be issued.

The @PS command allows the user to test the number of lines remaining on the current page. If there are not enough to satisfy the request the carriage is skipped to the

top of next page. If @PS is used within an incomplete line and the results forces a new page, the incomplete line will be printed on the next page.

Example:

@PG The top of the page

@PS008 Skips if there aren't 8 printable lines left on the current page.

Page extent

@HD controls the number of lines to be skipped from the top of a page before the text is printed. The number of lines to be printed on a page is set by the @BD command.

Example:

@HD002 ---> leaves five spaces at the top of each page since the printer hardware is usually set three lines from the top of the page.
@BD040 ---> allows 40 lines of printable text to appear on a page before going to the next page including the header spacing command.

LINE CONTROL

In the category of line control are those commands which control line justification and column spacing. Lines can be terminated immediately by the @DB command described in the section Line termination. Text flow within a line is bracketed by the left and right margins set by the @LM and @RM commands.

Example:

```
@LM040@RM050   @DB/hello   /i   am text to be
                                    formatted.
/if_you_run_me_together_/i_will_still_print_out_cor
rectly.

                        Hello I am
                        text to be
                        formatted.
```

If_you_run_
me_together
_I_will_sti
ll_print_ou
t_correctly
.

Line justification

Four modes of line justification are available: @LJ, left justification (as internally generated); @RJ, right justification; @CN which centers lines between the margins; and @EM which right and left justifies (evens the margins). Even margins are assumed when FMS is initially used.

These commands go into effect when issued and remain in effect until either a Reset or another line justification command is issued.

Example:

centering looks like this

right
justification

left
justification

even margins look like this

Only information between the left and right margins are affected by line justification.

Centering is performed by computing the location of the centeral point between the left and right margins and subtracting half the length of the text in the main line to compute the starting output location of the text in the main line.

Even Margins are obtained by evenly distributing the blanks left after the last character in the main line, up to the right margin, between the words in the line group. The direction of distribution is switched before each line is started to insure uniform printed page density (roughly the same method used in TEXT/360).

Underlining

The @UN command is provided for underlining main line text. Its effect is terminated by the @NU command. The underline character can be changed by the @UL command for special overprinting effects. The overprint line is used for storing the underline characters for printing.

Example:

```
This @UNword@NU is underlined. ---> This word
is undelined
```

Boldfacing

The @BF commands will cause the text following to be overprinted, giving a darker imprint to the text. This may be turned off with the @NF command. It is useful for emphasizing words or titles.

Example:

```
@PPnormal @BFbold@NF normal ---> normal bold
normal
```

Paragraph indentation

The paragraph indentation is set by the @IS command. An unusual and effective technique for page formatting is to set the paragraph indentation to the left of the left margin so that the initial line of subsequent new paragraphs overhang the left margin [see the general format of Appendix I].

Example:

```
@IS040@LM045@RM055
@PPx x x x x x x x x x x x x x @PPy y y y y  y
y y @PPz z z z z z@DB --->
                        x x x x x x x x
                             x  x x x x
                             x x
                        y y y y y y y y
```

z z z z z z

tabbing

Standard typewriter relative position tabbing is performed by the @TB command. These tab stops can be changed by the @ST tab list. An @AT starts the text at the exact location specified. Remember, when anything except left justification is in effect, the location of a given character may change at the time of the printing of the line.

Example:

```
@LJ
@PP@AT0601@AT0552@AT0503@DB --->
                                3    2    1

@PP1@TB2@TB3 --->
1  2    3
```

Column skipping

PMS normally leaves one space between words. To leave more than one space one of the following commands is used: @SS leaves a fixed number of spaces, @BS backspaces a fixed number of columns, @"" formats text without blank suppression until the next command is encountered. The space skipping commands do not blank out any existing characters in a line. Lines are blanked and assumed to be blank after each new line is started.

Example:

```
a   b   c ---> a b c
a@SS003b@SS003c@SS003 ---> a    b    c
@""a   b   c@DB ---> a    b    c
```

LINE SELECTION

FMS utilizes a multiple line format for handling subscripts, superscripts, boldface, and overprinting. By specifying the line you wish to write in, and where in the line you want to write, the program will take care of line printing. An @SU command will use the superscript line; an @SB will use the subscript line; an @OP uses the overprint line; and the @ML and @MX commands return control to the main line, at either the current location, or the last used position. The boldface line is set by a switch and is filled with main line data automatically.

As soon as either an @SU or @SB is issued the system changes to a special mode that leaves extra blank lines for unused subscript and superscript lines. This mode can be disabled by doing an @NS to directly turn off the mode or @RS to restore initial conditions. If a whole report is to be done in this way it is advisable to issue a dummy @SU or @SB command to set this mode at the very start of the program. The output generated looks best printed at 8 or 10 lines/inch. This mode can be suppressed with the @NS command.

Example:

```
  @PP    e@SUsin(x)@ML    +    y@SBij@MX@SUcos(z)@ML=
3.68@OP//@MX0@DB@NS
```

$$e^{\sin(x)} + y_{ij}^{\cos(z)} = 3.68$$

NOTE: If a line with subscripts and superscripts is in progress when a mode reset is issued the subscript and superscript lines are not printed even though there may be data in their buffers. The line group must be dumped before the sub-superscript mode suppress is issued.

PROGRAM STATUS GROUP CONTROL

@S# saves all the variables in the current PSG; @R# uses a PSG which has previously been saved. An @RS restores the PSG to its default value. This series of controls is very useful for complex format control. "#" is replaced by a save area number from 0 to 4.

Example:

```
  @PPhello @S4  @IS050  @PPhello  @R4  @PPhello@DB
--->
  hello
                                        hello
  hello
```

III. FMS BEGINNERS COOK BOOK

High quality reports can be generated in FMS with three and a half FMS commands:

@PP to go to a new paragraph,

@MU1,2, and 3 ... @MR predefined macros for setting up main and subordinate topics,and

/ for capitalization.

EXAMPLE:

```
@MU1THE NAME OF A MAJOR TOPIC @MR
@MU2TOPIC ONE@MR
@PP/THIS IS THE FIRST PARAGRAPH OF THE FIRST TOPIC. /THE
TEXT CAN BE TYPED IN WITH AS MANY INTERSPERSED BLANKS
AS YOU WISH SINCE THEY WILL ALL BE CORRECTLY ADJUSTED BY
/F/M/S.
@PP/PARAGRAPH TWO HERE.
@MU2TOPIC NUMBER TWO@MR
@PP/PARAGRAPH ONE OF TOPIC TWO.
@MU3SUB-TOPIC UNDER TOPIC TWO@MR
```

The next page is the result of running this example under FMS.

THE NAME OF A MAJOR TOPIC

TOPIC ONE

This is the first paragraph of the first topic. The text can be typed in with as many interspersed blanks as you wish since they will all be correctly adjusted by FMS.

Paragraph two here.

TOPIC NUMBER TWO

Paragraph one of topic two.

sub-topic under topic two

IV. MACROS

Macros permit the invocation of a number of difficult format control sequences as a single command. Valid FMS commands, text, or combination of the two can be saved in a macro and inserted into the input stream whenever needed.

Macros are controlled by seven special commands: macro define, macro end, macro use, macro revert, macro clear, macro on, and macro off.

The structure of the macro is stated in a macro define [@MD] including a one chararacter label that identifies the macro. It is terminated by a macro end [@ME] statement. To destroy an existing macro and replace it with another macro equal to or shorter than the original macros length, issue a define macro with the same name.

A macro is invoked by using its name in a macro use [@MU].

If the macro requires information from the source input before it has terminated, a macro revert [@MR] will cause the next text to be brought in from the previous input stream. Control is returned to the macro when a macro revert is encountered in the input stream. By issuing a macro use followed by many text fields delimited by macro reverts a dialoque is set up between the macro and the input stream.

It is important to note that an @MR in a macro implies that an @MR also exists in the input stream to return control to the macro for normal termination. A macro's control is finally released by exiting by way of the @ME terminal, not by an @MR.

To destroy all previously defined macros the macro clear [@MC] command is used.

The macro on and macro off [@MO and @MF] gives conditional control to macro definition and use. When a macro off is issued the macro in question is ignored whenever it is referenced until a macro on is issued. A macro on is issued automatically whenever a macro is defined or re-defined.

Nested macros are not yet fully supported.

EXAMPLE:

@MD?@S4@SK002@UN@UC@MR@LC@NU@SK001@R4@ME?

xxxxx@MU? stuff @MRyyyyy@DB
xxxxx

STUFF

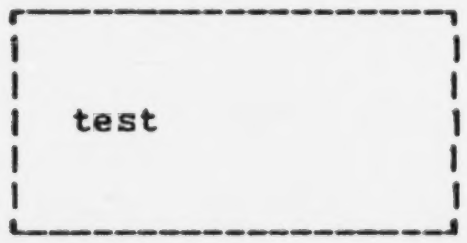
yyyyy
(notice the effect of the imbedded blanks)

The following complex set of commands define a box with a line of text inside:

@ST020040
@MDR@DB@TB/$@MUS/¬@MER [The top]

@ST020040
@MDR@DB@TB/$@MUS/¬@MER [The top]
@MDS@UC||||||||||||||||||||@LC@MES [The horizontal bar]
@MDT@DB@TB|@TB|@MET [The sides...no text]
@MDU@DB@TB|@SS002@MR@TB|@MEU [sides with a revert text request]
@MDV@DB@TB/_@MUS/,@MEV [The bottom]
@MDW@MUR@MUT@MUT@MUU@MUT@MUT@MUV@DB@MEW [The complete box]

@MUWtest@MR [Invoking the box]



DEFAULT MACROS

Three predefined macros are supplied when PMS is initially loaded (see appendix III) and were used to format the major and minor headings used in this manual.

V MODEL 20 TAPE FILE MANAGEMENT

FMS is capable of tape file management on the model 20 system. Any output directed to the printer can be simultaneously directed to a tape.

Tape output operations are initiated by an @FWnnn command which begins to write out information on the tape starting at the nnnth file.

Tape operations are terminated by executing an @FT which ends the current file. If the user is in tape write mode he is shifted back to read mode by ending the file explicitly, reading the last card from the card reader, or starting to print a file.

The @FPnnn command prints the nnnth file on the tape. Since the tape was treated like a duplicate printer when writing the file, the printing of the file can be performed out of sequence and interspersed with card and macro input text without a break in continuity.

If you wish output being written on tape not to be printed the commands @ON or @OY will suppress or permit printing.

Example:

```
@ON @FW001
  Text input here is not printed on the printer
but is put on tape.
@OY
@FT
  This text input prints on the printer directly.
@FP001
  @|* the first file is printed @*|
```

RESULTS: The second input is printed first and the first input is printed second.

APPENDIX I COMMAND LIST

KEY TO SYNTAX

i|j|m|n.....Integers between 0 and 9, each letter representing one digit.

aaaa.....An alphameric field with possible case conversion.

xxxxx.....unconverted character input exactly as typed on the input media.

COMMANDS

Blanks following commands that cause exact column positioning will allow a possible single space variation.

Unless otherwise specified no integer may be less than 0 or greater than 132.

Command definitions followed by an "▪" are reserved for the unsupported model 20 implementation.

Command definitions followed by an "□" are not defined in the model 20 implementation.

@ALnnn □ Absolute Left margin
Define the left margin for purposes of page numbering only. It is initially set to the default value of the left margin and is not in the PSG. See also @AR.

@ARnnn □ Absolute Right margin
Define the right absolute margin for page numbering. It's initial value is equal to the default right margin. See also @AL.

@ATnnn Absolute Tab
Set the line location pointer to "nnn". No position testing is performed allowing text to be placed anywhere in the 132 positions available per line.

@BF □ Bold Face font
Over strike all non-blank characters in the main line following this command. This implementation over strikes three times.

@BDnnn Body length Definition
Set the total number of lines on a page less 3 (the usual paper setting in the printer is three lines from the top), starting from the top of the page. If the length of the page set is greater or equal to the current line number than a new page is started after the line group currently being generated is completed.

@BSnnn □ BackSpace
Backspace the line output pointer "nnn" positions. If the left margin is exceeded the line output pointer is set to the current value of the left margin.

@CAx □ Change the Attention character
Change the attention character (default "@") to the single character specified. A case change qualification is not accepted in this special case and the use of the case changed alternate for this character is lost.

@CCn Automatic Carriage Control
Insert n-1 blank lines between each line group printed. If "n" is equal to 0 then n=1 is assumed with the addition of an extra blank line before each new paragraph when single spacing text. This provides an improvement in legibility.

@CN — CeNter lines

Center between the right and left margin all lines printed after the execution of this command including any lines in an incomplete state. The margins used are those which were set at the time the line group in progress was begun. Centering assumes that the line initially started at the left margin.

@CQx ¤ — Case Qualification character change

Modify the case change qualifier (default "/"). The use of the case changed alternate for this character is lost.

@DB — Dump line group Buffer

Dump all text found in the line buffers observing all line justification conventions, with the exception of even margins, and normal carriage control is taken. If the line buffers are empty, no print action is taken. The line output pointer is subsequently set to the left margin constant.

@DE — Dump buffer allowing Even margins

Dump all text found in the line buffers in exactly the same manner as the @DB except that the Even Margin option, if set, is observed. This command might be useful if an absolute tab or new paragraph command was used for setting up marginal notes while text between the margins is required to be right and left justified in order to blend with the rest of the text.

@EM — Even Margins

Simultaneously right and left justify, by inserting extra blanks between words, all text found between the right and left margins. The direction of blank insertions is switched before each line is generated to balance the overall page print density. Text not between the left and right margins is not affected by justification.

@FPnnn ¤ — File Print

Print completely the contents of file "nnn". The line group will be dumped and if the last tape operation was a file print data transfer to tape will stop. If the last operation was a file write the file is ended and data transfer will stop.

@FT - File Terminate
Terminate the current tape file imediately and position to the beginning of the next tape file.

@FWnnn - File Write
Copy onto tape any output directed to the printer after the execution of this command starting at file "nnn". Any files greater than "nnn" existing on the tape will be destroyed by this operation.

@HDnnn Set the HeaDer spacing
Set the total number of lines (less three) to be skipped at the top of each new page. Since the IBM 1403 printer is usually set to start printing three lines from the top of a page these three lines must be taken into consideration when setting the header.

@ISnnn Indentation Set
Set the paragraph indentation to "nnn". If it is equal to the left margin, one blank line is inserted before a paragraph conforming to standard block paragraphing conventions.

@LC Lower Case
Convert to lower case all text following this command before printing. This is used to terminate an @UC command.

@LJ Left Justify
Left justify all line groups (lines are left unchanged in the line group buffers). Before special justification techniques are applied all lines have this format before printing.

@LMnnn Left Margin set
Set the left margin to "nnn". This is not in effect until the current line group is dumped.

@MC Macro directory Clear
Clear all macro descriptions enabling new ones to be specified.

@MDa — Macro Definition
Define the start and name of a new macro. All instructions and text following this instruction, until an @MEa command, are stored as a macro named "a".

@MEa — Macro definition End
Terminate the scope of an @MDa command. This command must be preceded by an @MDa command or an invalid command message will be issued.

@MFa — Macro oFf
Declare macro "a" to be temporarily non-existent. This command can be nullified by the @MOa command or by re-defining the macro. The macro itself is not changed by this command.

@ML — Return to current relative position in the Main Line
Output all the following text on the main line starting at a point immediately following the last character generated. This is a possible termination for the @SU, @SB, or @OP commands. See also @MX.

@MOa — Macro On
Declare a macro as existing after an @MFa has been issued. This function is performed automatically by an @MDa.

@MR — Macro Revert back to the last used input stream (macro or card)
Permit a macro to release control to the card reader input stream and obtain control again. This is used to supply parameters to a macro.

@MUa — Use a previously defined Macro
Insert all the text in macro "a" into the input stream at this point.

@MX — Return to Main line to eXactly the last used position in main line
Place all text following this command on the main line, starting after the last used position in the main line. See also @ML.

@NF ¤ — Normal Face font
Resume normal single strike printing of the main line. This nullifies the effect of @BF.

@NU — No Underlining
Terminate underlining.

@NS ¤ — No Sub or superscripting
Turn off super- subscript mode. Be sure that the line buffers are completely empty before issuing this command or incompletely generated subscript and superscript lines will not be printed.

@ON — Output is Not printed on the printer
Do not print any output directed to the printer after the execution of this command. A line group in progress will therefore not be printed unless it is dumped before the command is issued.

@OP — OverPrint
Save the location of the line output pointer and insert text in the overprint line.

@OY — Output is permitted to print
Restart printer output by cancelling the @ON command. The contents of the line group in progress will not be printed unless the line group is dumped before printing is enabled.

@PBnnn ¤ — Page number at the Bottom of the page
Generate the page number "nnn" centered between the absolute left and right margins two lines below the last printable line on the page as specified by the @BD command. See also @PN.

@PG — New PaGe
Dump the last line group, start a new page, and establish a new header displacement. Top of page numbering, if requested, is performed automatically.

@PNnnn top of Page Numbering
Generate the current page as "nnn" right justified on the first printable line of a page, before header spacing is taken. Since FMS generates output one line at a time this form of page numbering will not be able to set a proper page number until the next page is started. Page numbers are incremented automatically. If "nnn" is equal to "***" then page numbering will be suppressed.

@PP New ParagraPh
Dump the line buffers as in a @DB and the output position is set to the paragraph indentation constant. If the paragraph indentation is equal to the left margin or if @CC0 {single space, special option} has been specified, an extra normal carriage control is performed.

@PSnnn Conditional Page Skip
Skip to the next page if "nnn" lines can not be placed on this page. The line group in progress at the time of execution of this command is not dumped.

@Rn Reset PSG using reset area n.
Restore all the format description elements described in Appendix III to what they were at the time of the last @Sn. If the nth reset area has not been set this instruction is ignored. "n" must be between 0 and 4.

@RJ Right Justify
Right justify All lines following the execution of this command.

@RMnnn Right Margin set
Set the right margin to "nnn". As in @LM this command is not in effect until the current line group is dumped.

@RS ReSet to default PSG {Program Status Group}
Reset all pertinent overall format variables to the original default values. All of them will not be in effect until the start of the next line. See also @Rn.

@Sn — Save the current PSG in reset area "n"
Save overall PSG format controls in save area "n". "n" must be between 0 and 4. See also @Rn.

@SB — Start suBscript
Save the location of the line output pointer and insert text in the Subscript line. See also @SU.

@SKnnn — SKip lines over page boundaries
Dump the line buffers, including carriage control, and insert "nnn" lines without regard to page boundaries.

@SPnnn — Skip lines or new Page
Dump line buffers including carriage control and insert "nnn" lines. If the skip goes over a page boundary skipping stops at the top of the next page after header spacing.

@SSnnn — Skip Spaces
Add "nnn" to the current value of the line output pointer. If the resulting location is greater than the right margin then the line output pointer is set equal to the right margin.

@STiiijjj... mmm aaa — Set Tab list
Set up to fifteen relative tab positions to the values specified. Values should be in ascending order and the last value in the list must be followed by at least on blank. If the new tab list is shorter than fifteen values, only the first replaced tab positions are changed.

@SU — Start sUbscript
Save the line output pointer and insert all text in the superscript line. This mode may be suppressed by an @RS or @NS.

@TB — TaB (relative)
Set the line output pointer to the first value saved in the relative tab table that is greater than the current value of the line output pointer.

@UC — Upper Case
Convert all characters following this command to upper case before processing. @LC will reset back to lower case.

@ULa — Change the Under Line character
Replace the underline character with "a". This is useful for strike-out and other special effect text. The underline character "a" may be upper case (preceded by a "/").

@UN — Start UNderlining
Underline all non-blank characters placed in the main line following this command.

@"" — As is text
Dump the following text with blanks unsuppressed, returning to normal mode when the next control command is encountered (any control command).

@@@ — Output the character "@"
Print the current attention character.

@|* — Start comment.
Text following this command is not printed and the line output pointer remains unchanged. All commands except the @*| will not be executed.

@*| — End comment.
Resume normal un-commented text processing.

/a — Change print case
If PMS is in upper case mode the character "a" is made lower case. if in lower case mode the character is made upper case.

// — Output the character /

APPENDIX II CHARACTER SET

input	@LC	@UC
A	a	A
B	b	B
C	c	C
D	d	D
E	e	E
F	f	F
G	g	G
H	h	H
I	i	I
J	j	J
K	k	K
L	l	L
M	m	M
N	n	N
O	o	O
P	p	P
Q	q	Q
R	r	R
S	s	S
T	t	T
U	u	U
V	v	V
W	w	W
X	x	X
Y	y	Y
Z	z	Z
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

input	@LC	@UC
$	$	↱
<	<	{
>	>	}
%	%	°
+	+	±
*	*	■
(	(	(
)	)	)
_	_	↳
-	-	–
=	=	≠
#	#	□
¬	¬	↴
|	|	—
'	'	[
&	&	†
,	,	↲
;	;	≥
.	.	•
:	:	≤
?	?	!
"	"	]
¢	¢	+

APPENDIX III DEFAULT PSG FORMAT CONDITIONS

---------- in PSG ----------

Subscript and superscript	OFF
Case	LOWER
Tabs	20 25 30 35 40 45 50 55 60 65 70 75 80 85 90
Body length	56 lines
Header spacing	3 lines {plus 3 for printer alignment}
Paragraph indentation at location	20
Right margin	75
Left margin	15
Underlining character	_
Carriage Control spacing	0 {see appendix I}
Blank suppression (quote mode)	ON
Underlining	OFF
Bold face	OFF
Line justification	Even Margin

---------- not in PSG ----------

@MD1	@RS@PG@SK004@UC@CN@MR @LC@DB@EM@SK002
@MD2	@SK002@PS010@UC@UN@MR @LC@NU@SK001
@MD3	@SK001@PS010@UN@MR@NU @SK001

APPENDIX IV ERROR MESSAGES

When FMS makes a decision to ignore a request made by the user, it issues a message that includes the last analyzed instruction and an error code indicating what occured. The formatted line containing the error is printed somewhere after the error message is issued since the errors were detected by the line group in progress. Processing will always continue.

**********ERROR CODE=1 , LAST INSTRUCTION = 'XX '**********

Error Code	Explanation
01	The command used was not a valid FMS instruction.
02	A non-numeric character has been found in a numeric field.
03	A numerical constant is greater than 132.
04	An attempt has been made to overflow the relative tab position table.
05	Too many macros are in effect.
06	This macro is undefined.
07	A macro revert has been issued before the macro input stream has been used.
08	Macro storage is full.
09	There is no room left in the macro directory.

10 A non-numeric character has been used in a page number command.

11 An attempt has been made to tab off the end of the tab table.

12 An attempt has been made to use a PSG that has not been set.

13 The length of the macro just defined is greater than the macro it replaces. The next macro in storage has been partially destroyed.

14 An attempt has been made to set the left margin equal or greater than the right margin.

15 An attempt has been made to set the right margin equal or less than than the left margin.

16 An @ME has no @MD preceding it.

17 A tape input/output operation has failed.

APPENDIX V MODEL 20 HARDWARE CONSIDERATIONS AND OPERATION

MINIMUM SYSTEM REQUIREMENTS

8K (limited version) or 12K (full version) CPU, card reader, printer, tapes (full version only, unit 081).

LOADING FMS

FMS is loaded in the primary hopper of the IBM 2560 MFCM. The address switches are set to "009C". Press "check reset" and then "load". The program will halt with "0FF" in the load lights.

In the event of a read check the card in error is the second card in the first stacker after MPRO has been pressed.

NOTE: The printer should be aligned to start printing three lines from the top of a page.

HALT CODES

222 Card reader error

0FF Program ready to start its first run; press start to continue.

1FF Program ready to continue after last card.

2FF Mount a tape, place it in ready status, press start on the CPU.

APPENDIX VI EXAMPLES OF FMS

Sangren, Ward Conrad, 1923-

Digital computer and nuclear reactor calculations. New York, Wiley [1960]

208p. illus. 14cm.

Includes bibliography.

1. Electronic Digital Computers 2. Nuclear reactors--Tables, calculations, etc. I. Title

QA76.S3 510.78 60-11728

Electronic Digital Computers

Sangren, Ward Conrad, 1923-

Digital computer and nuclear reactor calculations. New York, Wiley [1960]

208p. illus. 14cm.

Includes bibliography.

1. Electronic Digital Computers 2. Nuclear reactors--Tables, calculations, etc. I. Title

QA76.S3 510.78 60-11728

Nuclear reactors--Tables, calculations, etc.

Sanqren, Ward Conrad, 1923-

Diqital computer and nuclear reactor calculations. New York, Wiley [1960]

208p. illus. 14cm.

Includes bibliography.

1.Electronic Digital Computers 2.Nuclear reactors--Tables, calculations, etc. I. Title

QA76.S3 510.78 60-11728

Extract from the Campus Computing Network Users Manual.

4.3 Assembler Language

4.3.1 Description of the Language

The IBM OS/360 Assembler language is a symbolic programming language that permits the programmer to use all machine functions as if he were coding in 360 machine language.

The Assembler program that processes the language translates symbolic instructions into machine-language instructions, assigns storage locations, and performs auxiliary functions necessary to produce an executable machine-language program.

4.3.2 Assembler Language Documentation

IBM System/360 Operating System

Assembler Language, Form C28-6514

This publication contains specifications for the Operating System Assembler language. Part I of the manual describes the Assembler language, and part II describes an extention of the Assembler language -- the Macro Language -- used to define macro instructions.

IBM System/360 Operating System

Assembler (F) Programmer's Guide, Form C26-3756

This publication complements the IBM/360 OS Assembler Language manual. It provides a guide to program assembling, linkage editing, executing, interpreting listings, and Assembler programming considerations. Included in Appendix A of the manual is a description of the Assebler language diagnostic messages.

IBM System/360 Principles of Operation, Form A22-6821

This publication is the machine reference manual for the IBM System/360. It provides a direct, comprehensive description of the system structure, arithmetic, logical branching, status switching,

input/output operations, and the interruption system. It is the only manual that gives a detailed description of the various machine instructions.

```
@MU1APPENDIX VI   EXAMPLES  OF EMS@MR  @MD=@D8
                  @LJ@AT015@MR@PP@MR@PP@MR@PP@MR@SK001@PP1.@MU: 2.@MU: /I.
 /TITLE@SK001@MR@S4@EM@DE@R4@SK001@ME=  @MD:/ELECTRONIC /DIGITAL /COMPUTERS@ME:
  @MD;/NUCLEAR REACTORS--/TABLES, CALCULATIONS, ETC.@ME:  @LM020@IS023@RM070
@DB@PS020
@MU=/SANGREN, /WARD /CONRAD, 1923-@MR/DIGITAL COMPUTER AND NUCLEAR REACTOR
 CALCULATIONS.  /NEW /YORK, /WILEY /'1960/"@MR208P. ILLUS.  14CM.@MR/INCLUDES
 BIBLIOGRAPHY.@MR/Q/A76./S3 510.78 60-11728@MR
@DB@PS020
@D8----------------------------------
@DB@AT015
@MU:
@D8
@MU=/SANGREN, /WARD /CONRAD, 1923-@MR/DIGITAL COMPUTER AND NUCLEAR REACTOR
 CALCULATIONS.  /NEW /YORK, /WILEY /'1960/"@MR208P. ILLUS.  14CM.@MR/INCLUDES
 BIBLIOGRAPHY.@MR/Q/A76./S3 510.78 60-11728@MR
@DB@PS020
@D8----------------------------------
@DB@AT015
@MU:
@D8
@MU=/SANGREN, /WARD /CONRAD, 1923-@MR/DIGITAL COMPUTER AND NUCLEAR REACTOR
 CALCULATIONS.  /NEW /YORK, /WILEY /'1960/"@MR208P. ILLUS.  14CM.@MR/INCLUDES
 BIBLIOGRAPHY.@MR/Q/A76./S3 510.78 60-11728@MR  @PG     @RS  /EXTRACT
 FROM THE /CAMPUS /COMPUTING /NETWORK /USERS /MANUAL.  @CC?
@DB4.3 @UN/ASSEMBLER /LANGUAGE@NU
@DB4.3.1 @UN/DESCRIPTION OF THE /LANGUAGE@NU  @SK001/THE /I/B/M /O/S//360
 /ASSEMBLER LANGUAGE IS A SYMBOLIC PROGRAMMING LANGUAGE THAT PERMITS THE
 PROGRAMMER TO USE ALL MACHINE FUNCTIONS AS IF HE WERE CODING  IN 360 MACHINE
 LANGUAGE.
@PP/THE /ASSEMBLER PROGRAM THAT PROCESSES THE LANGUAGE TRANSLATES SYMBOLIC
  INSTRUCTIONS INTO MACHINE-LANGUAGE INSTRUCTIONS, ASSIGNS STORAGE LOCATIONS,
 AND PERFORMS AUXILIARY FUNCTIONS NECESSARY TO PRODUCE AN EXECUTABLE MACHINE-LAN
GUAGE PROGRAM.
@DB4.3.2 @UN/ASSEMBLER /LANGUAGE /DOCUMENTATION@NU  @LM020@ST062055060065070
 @S1
@DB@UN/I/B/M /SYSTEM//360 /OPERATING /SYSTEM
@DB/ASSEMBLER /LANGUAGE@NU,@TB/FORM /C28-6514  @CC1  @SK001/THIS PUBLICATION
 CONTAINS SPECIFICATIONS FOR THE /OPERATING  /SYSTEM /ASSEMBLER LANGUAGE.
  /PART /I OF THE MANUAL DESCRIBES THE /ASSEMBLER  LANGUAGE, AND PART /I/I
 DESCRIBES AN EXTENTION OF THE /ASSEMBLER LANGUAGE  -- THE /MACRO /LANGUAGE
 -- USED TO DEFINE MACRO INSTRUCTIONS.  @R1
@DB@UN/I/B/M /SYSTEM//360 /OPERATING /SYSTEM
@DB/ASSEMBLER (/E) /PROGRAMMER'S /GUIDE@NU,@TB/FORM /C2A-3756  @CC1  @SK001/THIS
 PUBLICATION COMPLEMENTS THE /I/B/M//360 /O/S /ASSEMBLER  /LANGUAGE MANUAL.
  /IT PROVIDES A GUIDE  TO PROGRAM ASSEMBLING, LINKAGE  EDITING, EXECUTING,
 INTERPRETING LISTINGS, AND  /ASSEMBLER PROGRAMMING CONSIDERATIONS.  /INCLUDED
 IN /APPENDIX /A OF THE  MANUAL IS A DESCRIPTION OF THE /ASSEBLER LANGUAGE
 DIAGNOSTIC MESSAGES.  @R1
@DB@UN/I/B/M /SYSTEM//360 /PRINCIPLES OF /OPERATION@NU,@TB/FORM /A22-6821
  @CC1  @SK002/THIS PUBLICATION IS THE MACHINE REFERENCE MANUAL FOR THE
 /I/B/M  /SYSTEM//360.  /IT PROVIDES A DIRECT, COMPREHENSIVE DESCRIPTION
 OF THE SYSTEM STRUCTURE, ARITHMETIC, LOGICAL BRANCHING, STATUS SWITCHING,
 INPUT//OUTPUT OPERATIONS, AND THE INTERRUPTION SYSTEM.  /IT IS THE ONLY
 MANUAL THAT GIVES A DETAILED DESCRIPTION OF THE VARIOUS MACHINE INSTRUCTIONS.
```

APPENDIX VII JOB CONTROL LANGUAGE AT UCLA

THE STANDARD FORMAT FOR AN FMS RUN IS AS FOLLOWS:

```
//      JOB
//FMS           EXEC PGM=FMS
//STEPLIB       DD   DISP=SHR,DSN=CSW015.SSS.LIBRARY
//SYSPRINT      DD   SYSOUT=A,SPACE=(TRK,100,RLSE)
//SYSIN         DD   DATA

-------------- FMS PROGRAM --------------

/*
//
```

FOR OFF-LINE TAPE GENERATION ONLY THE SYSPRINT DD CARD NEED BE CHANGED

```
//SYSPRINT      DD   DISP=OLD,VOL=SER=OUTPUT,UNIT=TAPE9,
//                   LABEL=(1,BLP)
```

Ask the operator to pre-mount your tape named "OUTPUT".

Certain DCB parameters may be specified:

1 The default BLKSIZE (3458) may be changed to any multiple of 133.

2 The RECFM may specify either F (implying FBA) or V (implying VBA). For debugging purposes V is recommended because it is less expensive to use. For the final run F is required because VB printing on the model 20 is not yet supported. RECFM=F is assumed if nothing is specified.

Special PARMs are also available on the 91 system:

'I' intersperses source input data with final output.
'D' overprints all capital letters.
'C' dumps an abreviated synopsis of FMS commands executed, in the far right columns of the output page.
'L' accepts lower case input data without conversion.

APPENDIX VIII SPEED AND COST

Model 20 time is estimated at 20 dollars/hour real time.

Model 91 time is base on a special charging method in use at UCLA.

The cost of a final upper and lower case output run, using the 91 to generate a tape and having the tape printed on the 20, will usually be under 8 cents per page not including initial keypunching costs.

A run for error correcting purposes, in upper case only on the 91, is usually below 3.5 cents per page.

20 Card to Printer (TN chain)	16 cents/page
20 Card to Tape (no printer)	15 cents/page
20 Tape to Printer (TN chain)	5 cents/page
20 Card to Printer (PN chain)	15 cents/page
91 Card to Printer (PN chain only)	3.5 cents/page
91 Card to Tape (no printer)	3.5 cents/page
91 Tape to Printer (model 20 tN)	4 cents/page